TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

**------------------------------**

****

**MÔN HỌC: ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**

**Đề tài: Tìm hiểu về mô hình phát triển phần mềm Agile/Scrum**

***Giảng viên hướng dẫn***: **ThS. Trần Thị Hồng Oanh**

***Sinh viên thực hiện*:**

**Tống Quang Anh - 1810A01**

**Phí Xuân Diễn - 1810A01**

**Phạm Hồng Đức - 1810A03**

**Nguyễn Thành Hiếu-1810A02**

**Phạm Mạnh Hùng - 1710A04**

**Hồ Long Nhật - 1810A03**

**Vũ Huy Thiện - 1810A05**

**Đoàn Thị Tuyết - 1810A01**

**Hà Nội - 2021**

**MỤC LỤC**

[I. Tổng quan về quy trình phát triển phần mềm 3](#_heading=h.1fob9te)

[1. Các giai đoạn trong quy trình phát triển phần mềm 3](#_heading=h.3znysh7)

[2. Các mô hình phát triển phần mềm phổ biến hiện nay 4](#_heading=h.2et92p0)

[II. Mô hình Agile 8](#_heading=h.tyjcwt)

[1. Giới thiệu về mô hình Agile 8](#_heading=h.3dy6vkm)

[2. Đặc trưng của Agile 11](#_heading=h.1t3h5sf)

[3. Phương pháp(quy trình) Agile phổ biến 13](#_heading=h.4d34og8)

[III. Quy trình Scrum 19](#_heading=h.2s8eyo1)

[1. Giới thiệu về Scrum 19](#_heading=h.17dp8vu)

[2. Quy trình vận hành Scrum trong phát triển phần mềm 20](#_heading=h.3rdcrjn)

[3. Lợi ích của quy trình Scrum 23](#_heading=h.26in1rg)

[IV. So sánh giữa mô hình Agile/Scrum với mô hình truyền thống 30](#_heading=h.1ci93xb)

[V. Áp dụng mô hình Agile/Scrum vào tổ chức, doanh nghiệp(Hiếu) 32](#_heading=h.3whwml4)

[Tài liệu tham khảo 37](#_heading=h.2bn6wsx)

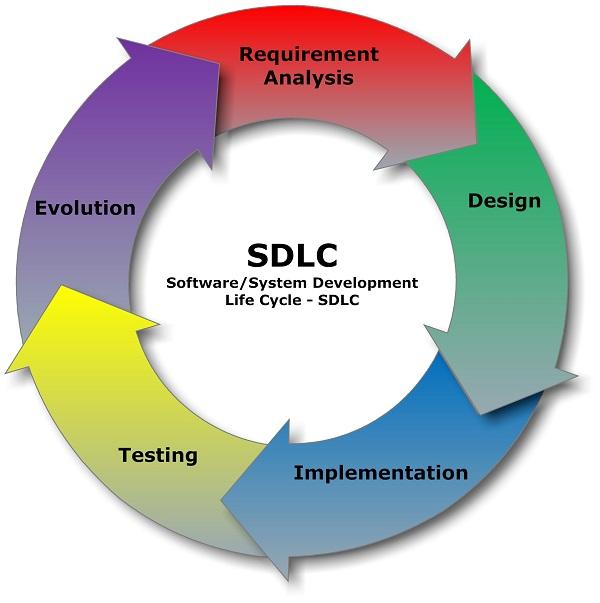
**Phân công thực hiện**

| **Họ và Tên** | **Nội dung** | **Ghi chú** |
| --- | --- | --- |
| Đoàn Thị Tuyết | - Tổng quan về quy trình phát triển phần mềm  - So sánh giữa mô hình Agile/Scrum với mô hình truyền thống |  |
| Phạm Mạnh Hùng | Giới thiệu về mô hình Agile |  |
| Phạm Hồng Đức | Đặc trưng của Agile |  |
| Vũ Huy Thiện | Phương pháp Agile phổ biến |  |
| Tống Quang Anh | Giới thiệu về quy trình Scrum |  |
| Phí Xuân Diễn | Quy trình vận hành Scrum trong phát triển phần mềm |  |
| Hồ Long Nhật | Lợi ích của quy trìnhScrum |  |
| Nguyễn Thành Hiếu | [Áp dụng mô hình Agile/Scrum vào tổ chức, doanh nghiệp](#_heading=h.qsh70q) |  |

## Tổng quan về quy trình phát triển phần mềm

### Các giai đoạn trong quy trình phát triển phần mềm

Quy trình phát triển phần mềm là một cấu trúc bao gồm tập hợp các hoạt động của tổ chức trong phát triển để sản xuất ra một sản phẩm phần mềm. Nó giúp cho mọi thành viên trong dự án từ người cũ đến mới, trong hay ngoài công ty đều có thể xử lý đồng bộ những công việc tương ứng vị trí của mình thông qua một cách thức chung của công ty.



Thông thường có 5 bước trong một quy trình phát triển phần mềm:

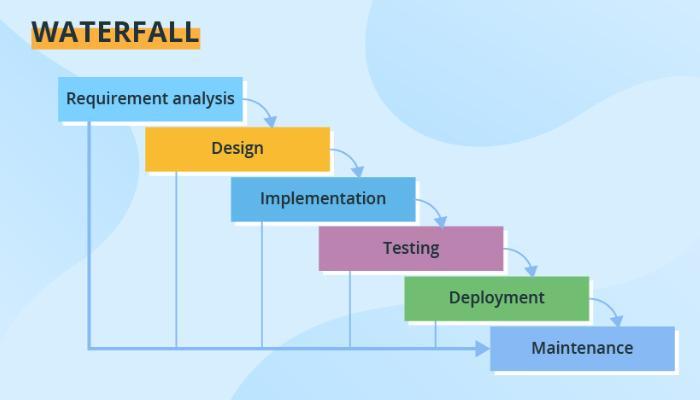
* **Yêu cầu và phân tích tính khả thi:** trong giai đoạn này, dự án được xác định chi tiết và việc phân tích tính khả thi của dự án được thực hiện. Để xây dựng một giải pháp khả thi, code sạch và thiết kế hấp dẫn là chưa đủ, trước tiên bạn cần nhóm phát triển hiểu sâu hơn về mục tiêu của dự án và thu thập đầy đủ tất cả các yêu cầu để phân tích.
* **Thiết kế:** thiết kế phần mềm là một bước sáng tạo của chu trình phát triển sản phẩm. Trong giai đoạn thiết kế, khái niệm thực tế của giải pháp được tạo ra, đó là kiến ​​trúc phần mềm chi tiết có đáp ứng các yêu cầu cụ thể của dự án hay không. Thiết kế phần mềm được tùy chỉnh bởi các kiến ​​trúc sư và kỹ sư phần mềm dựa trên các quy trình và tiêu chuẩn công việc nhất định và bao gồm thiết kế giải pháp / sản phẩm tổng thể rõ ràng cùng với cấu trúc và thiết kế cơ sở dữ liệu.
* **Phát triển và coding:** là viết mã và chuyển đổi tài liệu thiết kế thành phần mềm thực tế trong quá trình phát triển phần mềm. Giai đoạn này là dài nhất vì nó là xương sống của toàn bộ quy trình và bao gồm một số điều quan trọng cần chú ý.
* **Tích hợp và thử nghiệm:** Tùy thuộc vào các quy trình thử nghiệm được thông qua, nó có thể khác nhau. Tuy nhiên thông thường, các kỹ sư QA sử dụng các công cụ kiểm thử để thực hiện các bài kiểm tra về sự ổn định của phần mềm.
* **Triển khai và bảo trì:** Ứng dụng được chuyển sang sản xuất bao gồm truyền dữ liệu và thành phần trong khi trong các phiên bản ra mắt tiếp theo, chỉ những thay đổi cụ thể mới được triển khai. Đồng thời sản phẩm cũng được nâng cấp, cải tiến và tinh chỉnh theo phản hồi thực tế về hiệu suất của nó.

### Các mô hình phát triển phần mềm phổ biến hiện nay

Mô hình phát triển phần mềm hay quy trình phát triển phần mềm xác định các giai đoạn trong xây dựng phần mềm. Có nhiều loại mô hình phát triển phần mềm thường thấy sau đây.

**Mô hình thác nước - Waterfall model**

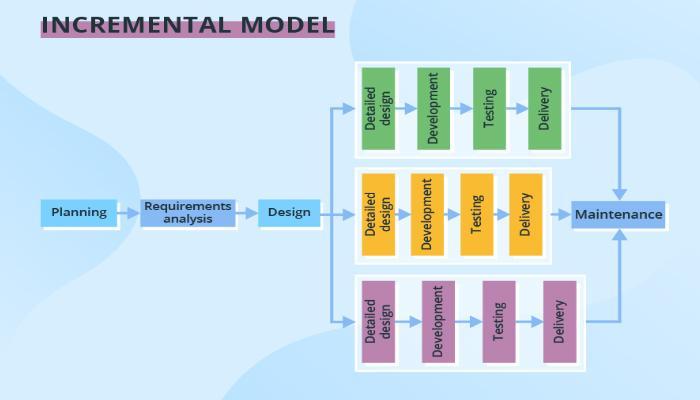
Là một mô hình SDLC theo tầng, trong đó quá trình phát triển giống như dòng chảy, áp dụng tuần tự các giai đoạn của phát triển phần mềm bao gồm phân tích, dự báo, hiện thực hóa, thử nghiệm, thực hiện và hỗ trợ. Quá trình này được thiết lập và theo dõi nghiêm ngặt và được xác định trước với các tính năng được mong đợi cho mọi giai đoạn của mô hình SDLC này.



Ưu điểm mô hình này là dễ phát triển và quản lý, tuy nhiên nó lại ít linh hoạt, rất khó để đo lường sự phát triển trong từng giai đoạn. Do đó, nó chỉ thích hợp những dự án ngắn hạn. Waterfall model là mô hình phát triển phần mềm đầu tiên được sử dụng.

**Mô hình tiếp cận lặp – Iterative model**

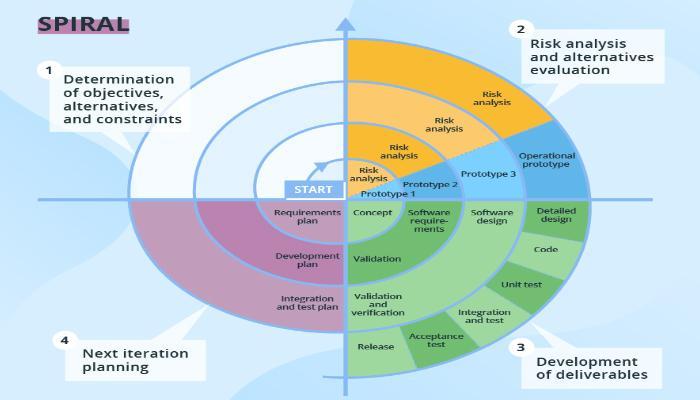
Mô hình này không cần đầy đủ các yêu cầu trước khi dự án bắt đầu, chỉ cần xác định yêu cầu chính là được. Quá trình phát triển có thể bắt đầu tùy theo những yêu cầu chức năng riêng biệt, có thể được mở rộng sau đó. Quá trình này lặp đi lặp lại, cho phép tạo ra các phiên bản mới của sản phẩm cho mọi chu kỳ. Mỗi lần lặp lại (kéo dài từ hai đến sáu tuần) bao gồm việc phát triển một thành phần riêng biệt của hệ thống và sau đó, thành phần này được thêm vào chức năng đã phát triển trước đó.



Ưu điểm của mô hình cuốn chiếu này là linh hoạt, thời gian làm tài liệu sẽ ít hơn so với thời gian thiết kế, dễ quản lý rủi ro. Tuy nhiên, điểm hạn chế của chúng chính là yêu cầu nhiều tài nguyên và phức tạp trong quản lý.

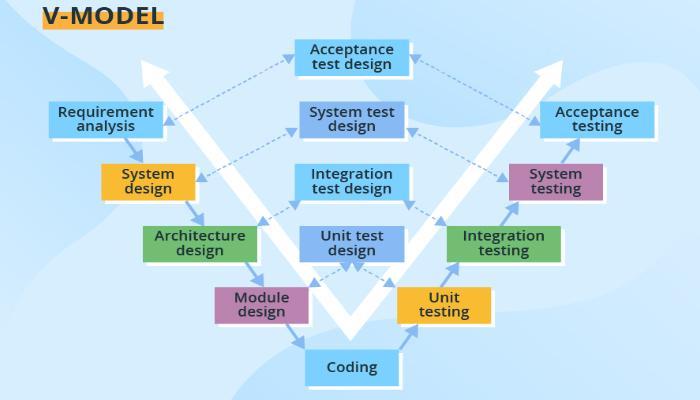
**Mô hình xoắn ốc – Spiral model**

Đây là mô hình phát triển phần mềm xoắn ốc có sự kết hợp giữa mô hình prototyping và mô hình thác nước, với điểm nhấn quan trọng là phân tích rủi ro.



**Mô hình V-shaped**

Mô hình phát triển phần mềm V-shaped hay còn gọi là mô hình chữ V là một bản nâng cao của mô hình thác nước cổ điển và dựa trên giai đoạn thử nghiệm liên quan cho mọi giai đoạn phát triển. Đây là một mô hình yêu cầu tính kỷ luật cao theo timeline và giai đoạn tiếp theo chỉ được bắt đầu sau giai đoạn trước.



Đây còn được gọi là mô hình “Xác thực và xác minh”. Mọi giai đoạn đều có kiểm soát quy trình hiện tại, để đảm bảo rằng việc chuyển đổi sang giai đoạn tiếp theo là có thể thực hiện được. Tuy nhiên, nó lại khó quản lý rủi ro, và không thích hợp các dự án thay đổi liên tục.

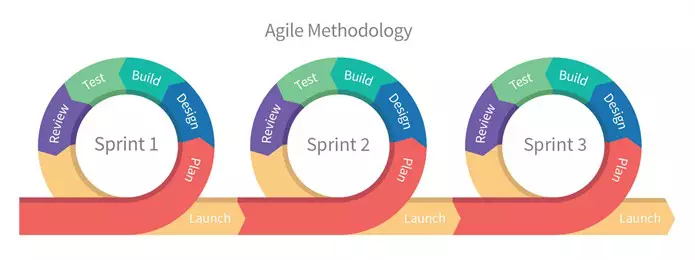
**Mô hình Agile**

Trong mô hình phát triển phần mềm này, sau mỗi lần lặp lại phát triển, khách hàng có thể nhìn thấy kết quả và để đưa nhận xét. Đây là một trong những ưu điểm của mô hình Agile. Cơ sở của mô hình này bao gồm các cuộc họp ngắn hàng tuần – Sprint là một phần của phương pháp tiếp cận Scrum.



## Mô hình Agile

### Giới thiệu về mô hình Agile

Trong các dự án, đặc biệt là các dự án phần mềm, có rất nhiều tham số thay đổi sự chính xác của yêu cầu sản phẩm như: Thời gian, công nghệ, xu hướng, ý tưởng, phản hồi, con người... và chúng ta đều không lường trước được, vì vậy mà Agile ra đời 

**a. Agile ra đời như thế nào**

* 2001: Một nhóm 17 người uy tín trong việc phát triển phần mềm đã gặp nhau. Họ đều là những đại diện cho những phương pháp phát triển riêng lẻ đang được thực hành lúc bấy giờ
* Cuối cùng họ đã đi đến thống nhất trong việc cho ra đời một bản tuyên ngôn Agile nhằm xác lập tinh thần và giá trị cốt lõi mà các phương pháp agile đều chia sẻ chung

**b. Định nghĩa**

* **Agile** là một phương pháp phát triển phần mềm linh hoạt, là một hướng tiếp cận cụ thể cho việc quản lý dự án phần mềm. Nó gồm một quá trình làm việc tương tác và tích hợp để có thể đưa sản phẩm đến tay người dùng càng nhanh càng tốt.

**c. Tuyên ngôn Agile**

Chúng tôi phát hiện ra cách phát triển phần mềm tốt hơn bằng cách thực hiện nó và giúp đỡ người khác thực hiện.

Qua công việc này, chúng tôi đã đi đến việc đánh giá cao:

* Cá nhân và sự tương tác - hơn là quy trình và công cụ
* Phần mềm chạy tốt - hơn là tài liệu đầy đủ
* Cộng tác với khách hàng - hơn là đàm phán hợp đồng
* Phản hồi với các thay đổi - hơn là bám sát kế hoạch

**Mặc dù các điều bên phải vẫn còn giá trị, nhưng chúng tôi đánh giá cao hơn các mục tiêu bên trái**

**1. Cá nhân và tương tác**

*Cá nhân và sự tương tác hơn là quy trình và công cụ*

* - Đây chính là nhân tố cốt lõi làm ra các sản phẩm trong đó
  + Tự tổ chức
  + Tự quản lý và duy trì động lực các cá nhân là những yếu tố then chốt cũng như việc phải đảm bảo một môi trường thuận lợi nhất cho quá trình tương tác giữa các cá nhân đó.

**2. Phần mềm chạy tốt**

*Phần mềm chạy tốt hơn là tài liệu đầy đủ*

* - Bởi vì đối với khách hàng, sản phẩm hoạt động tốt quan trọng hơn rất nhiều so với việc nhận được cả núi tài liệu liên quan đến sản phẩm nhưng lại không mang giá trị sử dụng ngay tức thì
* - Trong quy trình sản xuất cũ, thông thường chúng ta sẽ tập trung nhiều công sức cho giai đoạn nghiên cứu, phân tích, thiết kế, để có một kế hoạch chi tiết và đầy đủ trước khi bắt tay vào sản xuất -> điều này sau một thời gian khá lớn chúng ta sở hữu rất nhiều tài liệu phát triển thay vì sản phẩm thực tế cho người dùng

**Agile khuyến nghị chuyển giao sản phẩm trong giai đoạn đầu tiên**

**3. Cộng tác với khách hàng**

* Lý do là chúng ta muốn chắc chắn khách hàng muốn sản phẩm như thế nào và những tính năng nào thực sự mang lại giá trị cho họ
* Lúc đầu rất khó để liệt kê chi tiết và chính xác những tính năng này.

**Do vậy chúng ta rất cần khách hàng tham gia vào quá trình phát triển sản phẩm và đảm bảo những gì chúng ta đang làm là đúng ý muốn của họ, thật sự mang lại giá trị cho họ và tối ưu điều chỉnh sản phẩm để mang lại những giá trị này.**

**4. Phản hồi với các thay đổi**

*Phản hồi với các thay đổi hơn là bám sát kế hoạch*

* Giúp chúng ta nhanh chóng thích nghi với mọi yêu cầu mới, tối ưu hoá giá trị mang lại và giảm thiểu rủi ro

**Công cụ hỗ trợ**

* **Phần mềm họp trực tuyến** (ví dụ: GoToMeeting hoặc Google Hangouts): Đây là công cụ đầu tiên mà tất cả các nhóm Agile làm việc từ xa cần chuẩn bị.
* **Stories on Board:** Công cụ chạy trên web này hỗ trợ bạn và nhóm của mình trình bày hiểu biết chung về sản phẩm bằng cách xây dựng story map của sản phẩm.
* **Azure DevOps**: Chúng tôi sử dụng bảng Kanban – một công cụ trực quan hóa công việc và quy trình làm việc, để dễ dàng theo dõi và sắp xếp công việc trong nhóm sao cho đúng trình tự.
* **Các công cụ lên kế hoạch kiểm thử phần mềm**: Những nhóm làm việc chung thường tiến hành kiểm thử chấp nhận của người dùng dạng Excel dành riêng cho UAT. Điều này cho phép tester làm việc theo phần mềm quen thuộc để thực hiện kiểm thử và trả về kết quả.
* **AWS và Microsoft Azure Cloud:** Các đội ngũ phát triển phần mềm cần một số môi trường điện toán để phát triển và kiểm thử; Một cho phát triển, một cho thử nghiệm và một cho sản xuất.
* **Slack:** dùng để trao đổi công việc hàng ngày: Các nhóm làm việc từ xa có thể sử dụng Slack như một luồng liên lạc giữa các thành viên trong nhóm.
* **Planning Poker:** Để ước tính hầu hết các user story, chúng tôi chơi các ván bài bằng phần mềm Planning Poker (PlanningPoker.com). Phần mềm này không khác gì phương pháp Planning Poker thông thường được dùng để trình bày, thảo luận và ước tính một story user.

### Đặc trưng của Agile

* 1. **Tính lặp(Iterative)**

Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại. Các phân đoạn (được gọi là Iteration hoặc Sprint) này thường có khung thời gian ngắn (từ một đến bốn tuần). Trong mỗi phân đoạn này, nhóm phát triển thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai, kiểm thử (với các mức độ khác nhau) để cho ra các phần nhỏ của sản phẩm. Các phương pháp agile thường phân chia mục tiêu thành các phần nhỏ với quá trình lập kế hoạch đơn giản và gọn nhẹ nhất có thể, và không thực hiện việc lập kế hoạch dài hạn.

* 1. **Tính tăng trưởng và tiến hóa (Incremental & Evolutionary)**

Cuối các phân đoạn, nhóm phát triển thường cho ra các phần nhỏ của sản phẩm cuối cùng. Các phần nhỏ này thường là đầy đủ, có khả năng chạy tốt, được kiểm thử cẩn thận và có thể sử dụng ngay (gọi là potentially shippable product increment of functionality). Theo thời gian, phân đoạn này tiếp nối phân đoạn kia, các phần chạy được này sẽ được tích lũy, lớn dần lên cho tới khi toàn bộ yêu cầu của khách hàng được thỏa mãn. Khác với mô hình phát triển Thác nước – vốn chỉ cho phép nhìn thấy toàn bộ các chức năng tại thời điểm kết thúc dự án, sản phẩm trong các dự án agile lớn dần lên theo thời gian, tiến hóa cho tới khi đạt được trạng thái đủ để phát hành.

* 1. **Tính thích nghi (adaptive)**

Do các phân đoạn chỉ kéo dài trong một khoảng thời gian ngắn, và việc lập kế hoạch cũng được điều chỉnh liên tục, nên các thay đổi trong quá trình phát triển (yêu cầu thay đổi, thay đổi công nghệ, thay đổi định hướng về mục tiêu v.v.) đều có thể được đáp ứng theo cách thích hợp . Ví dụ, trong Scrum – phương pháp phổ biến nhất hiện nay – trong khi nhóm phát triển sản xuất ra các gói phần mềm, khách hàng có thể đưa thêm các yêu cầu mới, chủ sản phẩm (Product Owner) có thể đánh giá các yêu cầu này và có thể đưa vào làm việc trong phân đoạn (được gọi là Sprint trong Scrum) tiếp theo. Theo đó, các quy trình agile thường thích ứng rất tốt với các thay đổi.

* 1. **Nhóm tự tổ chức và liên chức năng**

Cấu trúc nhóm agile thường là liên chức năng(cross-functionality) và tự tổ chức(self-organizing). Theo đó, các nhóm này tự thực hiện lấy việc phân công công việc mà không dựa trên các mô tả cứng về chức danh (title) hay làm việc dựa trên một sự phân cấp rõ ràng trong tổ chức. Các nhóm này cộng tác với nhau để ra quyết định, theo dõi tiến độ, giải quyết các vấn đề mà không chờ mệnh lệnh của các cấp quản lý. Họ không làm việc theo cơ chế “mệnh lệnh và kiểm soát” (command and control).

Nhóm tự tổ chức có nghĩa là nó đã đủ các kĩ năng (competency) cần thiết cho việc phát triển phần mềm, do vậy nó có thể được trao quyền để tự ra quyết định, tự quản lí và tổ chức lấy công việc của chính mình để đạt được hiệu quả cao nhất.

* 1. **Quản lý tiến trình thực nghiệm (Empirical Process Control)**

Các nhóm agile ra các quyết định dựa trên các dữ liệu thực tiễn thay vì tính toán lý thuyết hay các tiền giả định (prescription). Việc phân nhỏ dự án thành các phân đoạn ngắn góp phần gia tăng các điểm mốc để nhóm phát triển thu thập dữ kiện cho phép điều chỉnh các chiến lược phát triển của mình. Nói cách khác, Agile rút ngắn vòng đời phản hồi (short feedback life cycle) để dễ dàng thích nghi và gia tăng tính linh hoạt. Theo thời gian, các chiến lược này sẽ tiến gần đến trạng thái tối ưu, nhờ đó nhóm có thể kiểm soát được tiến trình, và nâng cao năng suất lao động.

* 1. **Giao tiếp trực diện (face-to-face communication)**

Agile không phản đối việc tài liệu hóa, nhưng đánh giá cao hơn việc giao tiếp trực diện thay vì thông qua giấy tờ. Agile khuyến khích nhóm phát triển trực tiếp nói chuyện để hiểu rõ hơn về cái khách hàng thực sự cần. Trong giao tiếp giữa nội bộ nhóm, Agile khuyến khích trực tiếp trao đổi và thống nhất với nhau về thiết kế của hệ thống và cùng nhau triển khai thành các chức năng theo yêu cầu.

* 1. **Phát triển dựa trên giá trị (value-based development)**

Một trong các nguyên tắc cơ bản của agile là “sản phẩm chạy tốt chính là thước đo của tiến độ”. Nhóm Agile thường cộng tác trực tiếp và thường xuyên với khách hàng để biết yêu cầu nào có độ ưu tiên cao hơn, mang lại giá trị hơn sớm nhất có thể cho dự án.

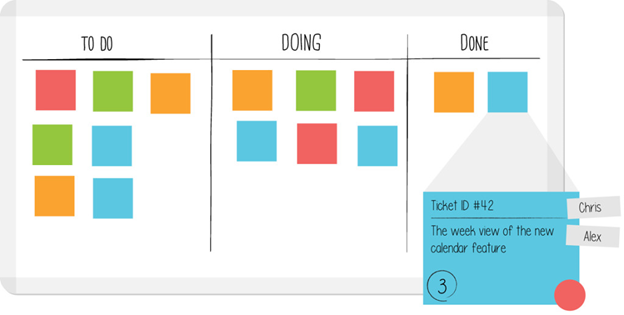
### Phương pháp(quy trình) Agile phổ biến

**a.Phương pháp Kanban**

**Khái niệm:**

Kanban là phương pháp Agile và nguồn gốc Kanban được phát triển vào cuối những năm 1940 bởi một kỹ sư người Nhật tên là Taiichi Ohno. Agile Kanban Framework tập trung vào việc trực quan hóa toàn bộ dự án trên các bảng nhằm tăng tính minh bạch của dự án và sự hợp tác giữa các thành viên trong nhóm.

Kanban là một phương pháp Agile nhưng không nhất thiết cần có tính lặp. Kanban cho phép phần mềm được phát triển trong một chu kỳ phát triển lớn. Mặc dù vậy, Kanban là một ví dụ về một phương pháp Agile vì nó đáp ứng tất cả mười hai nguyên tắc đằng sau tuyên ngôn Agile, bởi vì mặc dù nó không có tính lặp, nhưng vẫn có tính tăng trưởng.



**4 nguyên lý của Kanban**

* *Trực quan hóa công việc*

Bảng Kanban là công cụ để trực quan hóa công việc. Bảng Kanban bao gồm các cột tương ứng với trạng thái của công việc và thẻ Kanban (Thẻ Kanban là một hình ảnh đại diện cho một hạng mục công việc. Thẻ Kanban chứa thông tin có giá trị về nhiệm vụ và trạng thái của nó, chẳng hạn như tóm tắt về nhiệm vụ, người chịu trách nhiệm, thời hạn, v.v.). Mỗi công việc khi ở trạng thái nào thì được đặt ở cột tương ứng. Chúng ta có thể dùng một bảng vật lý hoặc một phần mềm hỗ trợ Kanban như Trello.

* *Giới hạn công việc đang làm (Limit WIP – Limit Work In Progress)*

Số lượng công việc đang được làm đồng thời ở mỗi trạng thái cần được giới hạn. Nguyên lý này giúp giới hạn những việc chưa hoàn thành trong tiến trình, từ đó giảm thời gian mỗi công việc đi qua hệ thống Kanban. Nguyên lý giới hạn WIP còn giúp cho nhóm làm việc tập trung, tránh lãng phí do phải việc chuyển qua lại giữa các công việc khác nhau.

* *Tập trung vào luồng làm việc*

Việc áp dụng nguyên lý giới hạn WIP và phát triển những chính sách hướng theo nhóm giúp nhóm có thể tối ưu hóa hệ thống Kanban để cải tiến luồng làm việc trơn chu.

* *Cải tiến liên tục*

Nhóm đo mức độ hiệu quả bằng cách theo dõi chất lượng, thời gian làm sản phẩm, v.v. để từ đó có những phân tích, thử nghiệm để thay đổi hệ thống nhằm tăng tính hiệu quả của nhóm

**Lợi ích của Kanban**

* *Lập kế hoạch linh hoạt*

Một nhóm Kanban sẽ chỉ tập trung vào công việc đang được tiến hành. Sau khi nhóm hoàn thành một hạng mục công việc, họ sẽ loại bỏ hạng mục công việc tiếp theo vào phần công việc đang làm. Chủ sở hữu sản phẩm có thể tự do sắp xếp lại công việc đang tồn đọng mà không làm gián đoạn nhóm vì bất kỳ thay đổi nào bên ngoài các hạng mục công việc hiện tại đều không ảnh hưởng đến nhóm.

* *Chu kì thời gian làm việc được rút ngắn lại*

Thời gian chu kỳ là lượng thời gian cần để một đơn vị công việc đi qua quy trình làm việc của nhóm – từ thời điểm công việc bắt đầu cho đến khi hoàn thành. Bằng cách tối ưu hóa thời gian chu kỳ, nhóm có thể tự tin dự báo việc phân phối công việc trong tương lai.

Trong Kanban, không phải mỗi người nắm giữ một kỹ năng, vì như vậy nếu người đó không hoàn thành tốt công việc thì sẽ sở thành điểm tắc nghẽn trong quy trình làm việc. Vì vậy, nhóm Kanban luôn hỗ trợ và bổ sung kỹ năng cho nhau, đảm bảo các thành viên luôn được học hỏi và không chỉ tập trung vào kỹ năng nào. Các kỹ năng được chia sẻ có nghĩa là các thành viên trong nhóm có thể đảm nhận công việc không đồng nhất, giúp tối ưu hóa hơn nữa thời gian chu kỳ.

* *Ít tắc nghẽn hơn*

Việc đa nhiệm đôi lúc có thể gây ra sự thiếu hiệu quả trong công việc do có quá nhiều đầu việc khác nhau trong nhóm. Đó là lý do tại sao một nguyên lý chính của Kanban là giới hạn số lượng công việc đang thực hiện (WIP). Giới hạn công việc đang tiến hành giúp ít tắc nghẽn và tăng dự phòng trong quy trình của nhóm do thiếu tập trung, con người hoặc kỹ năng.

* *Số liệu trực quan*

Một trong những giá trị cốt lõi là tập trung vào việc liên tục cải thiện hiệu suất và hiệu quả của nhóm với mỗi lần lặp lại công việc. Trong Kanban, công việc sẽ được theo dõi qua biểu đồ, biểu đồ này cung cấp một cơ chế trực quan cho các nhóm để đảm bảo rằng họ đang liên tục cải thiện. Khi nhóm có thể xem dữ liệu, sẽ dễ dàng phát hiện ra các điểm nghẽn trong quy trình (và loại bỏ chúng). Hai báo cáo phổ biến mà đội Kanban sử dụng là biểu đồ kiểm soát và sơ đồ luồng tích lũy.

* *Chuyển giao liên tục*

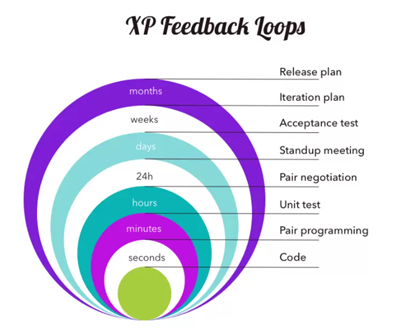
Chuyển giao liên tục (CD) là việc thường xuyên làm việc với khách hàng về tiến trình phát hành sản phẩm (đây là một đặc điểm nổi bật của Agile).

Tích hợp liên tục (CI) là thực hành tự động xây dựng và kiểm tra mã tăng dần trong ngày. Họ cùng nhau tạo thành một đường ống CI / CD cần thiết cho các nhóm phát triển (đặc biệt là cho các nhóm DevOps) để vận chuyển phần mềm nhanh hơn mà vẫn đảm bảo chất lượng cao.

**b. XP (Extreme Programming)**

**Khái niệm:**

*Extreme Programming* là một phương pháp phát triển phần mềm hướng đến việc nâng cao chất lượng phần mềm và khả năng đáp ứng với thay đổi yêu cầu người dùng. XP là một trong các phương pháp thuộc họ Agile, nó chủ trương đưa ra các bản phát hành thường xuyên thông qua các chu trình phát triển ngắn. Việc này là để nâng cao năng suất và tạo ra những thời điểm để tiếp nhận những yêu cầu người dùng mới. Tác giả của XP là Ken Beck, một kỹ sư phần mềm người Mỹ.



Tên gọi Extreme Programming (Lập trình Cực đoan) xuất phát từ ý tưởng phương pháp này đưa các hoạt động lập trình hữu ích lên một mức cao nhất có thể. Ví dụ, việc rà soát mã nguồn là một hoạt động hữu ích, và trong XP thì việc này được diễn ra liên tục, chẳng hạn là thông qua Lập trình cặp.

**Một số thành phần và đặc điểm của XP**:

Lập trình cặp, Rà soát mã nguồn, Kiểm thử đơn vị, Giữ mã nguồn đơn giản và rõ ràng, Sẵn sàng đón nhận các thay đổi, Trao đổi thường xuyên với khách hàng, Trao đổi thường xuyên giữa các nhà phát triển.

• **Các hoạt động trong XP:**

***Viết mã:*** XP coi mã nguồn là thành phần quan trọng nhất trong quá trình phát triển phần mềm. Không có mã nguồn thì không có sản phẩm chạy được.

***Kiểm thử***: XP nói về việc kiểm thử như sau: “Nếu một vài bài kiểm thử có thể loại bỏ được một ít lỗi thì nhiều bài kiểm thử sẽ loại bỏ được nhiều lỗi”. Do vậy XP khuyến khích việc tiến hành kiểm thử ở tất cả mọi mức độ. Từ Kiểm thử đơn vị cho đến Kiểm thử chấp nhận, Kiểm thử tích hợp,…

***Lắng nghe***: Các lập trình viên cần lắng nghe khách hàng của họ để biết được họ thực sự cần gì.

***Thiết kế***: Cố gắng thiết kế một kiến trúc cho phép loại bỏ phần lớn sự phụ thuộc trong hệ thống, cho phép thay đổi và mở rộng dễ dàng.

• **Các giá trị trong XP:**

***Giao tiếp***: XP chủ trương tạo ra một cách hiểu chung giữa các nhà phát triển về hệ thống mà họ đang xây dựng, tránh các hiểu lầm. XP khuyến khích thiết kế đơn giản, cộng tác giữa người dùng và lập trình viên, giao tiếp thường xuyên và phản hồi liên tục.

***Đơn giản***: XP khuyến khích việc bắt đầu với một giải pháp đơn giản nhất. Các tính năng khác được thêm vào khi cần thiết.

***Phản hồi:*** XP khuyến khích các phản hồi đa chiều:

* Phản hồi từ hệ thống: Thông qua việc kiển thử.
* Phản hồi từ khách hàng: Thông qua kiểm thử chấp nhận được viết bởi khách hàng và kiểm thử viên.
* Phản hồi từ nhóm: Nhóm ước tính mỗi khi có các hạng mục mới mà khách hàng muốn đưa vào.

***Tinh thần dũng cảm***: Tinh thần này được khuyến khích thông qua các hoạt động khác nhau.

***Tôn trọng***: Tôn trọng người khác và tôn trọng chính mình.

**c. CRYSTAL**

**Khái niệm:**

Người sáng lập ra Crystal là Alistair Cockburn. Theo ông, nó là “là một nhóm các phương pháp phát triển phần mềm, hoạt động bằng sức mạnh của con người, cực kỳ nhẹ và co giãn để phù hợp”. Về cơ bản, Cockburn tin rằng chính tài năng và cách tương tác giữa các thành viên trong nhóm sẽ mang lại lợi ích cho cả dự án.

Đây thực ra là một nhóm các phương pháp Agile gọi chung là Crystal (Pha lê). Chúng bao gồm các biến thể như: Crystal Clear (tối đa một nhóm 8 người), Crystal Yellow (tối đa một nhóm 10 đến 20 người), Crystal Orange (tối đa một nhóm 20 đến 50 người), Crystal Red (dành cho các nhóm lớn từ 50 đến 1000 người).

**Nguyên tắc của Crystal:**

Phương pháp Crystal tập trung vào các nguyên tắc như: Con người (People), Interactions (Sự tương tác), Community (Cộng đồng), Skills (Kỹ năng), Talent (Tài năng) và Communication (Giao tiếp). Mục đích của Crystal là đưa ra một quy trình phát triển phần mềm tốt nhất có thể. Theo đó, vấn đề cốt lõi phải là sự tương tác và cộng sinh giữa những người liên quan đến dự án.

**Ưu nhược điểm:**

*Điểm mạnh của Crystal bao gồm:*

* + Cho phép các nhóm làm việc theo cách mà họ cho là hiệu quả nhất
  + Tạo điều kiện cho giao tiếp nhóm trực tiếp, tính minh bạch và trách nhiệm giải trình
  + Phương pháp thích ứng cho phép các nhóm phản ứng tốt với các yêu cầu thay đổi

*Điểm yếu của Crystal bao gồm:*

* + Thiếu kế hoạch được hoạch định trước có thể dẫn đến sai lầm
  + Thiếu tài liệu có thể dẫn đến nhầm lẫn

Ngoài các phương pháp kể trên thì Scrum là phương pháp Scrum là phương pháp phổ biến nhất trong Agile

## Quy trình Scrum

### Giới thiệu về Scrum

Scrum là một framework quản lý dự án theo nguyên lý Agile. Framework Scrum được đội ngũ dùng để phát triển phần mềm, vận chuyển, và duy trì những sản phẩm phức tạp. Chúng ta có thể thường được nghe nói nhiều về Scrum và Agile nhưng thực ra chúng lại có phần khác nhau tương đối. Cốt lõi của Scrum là liên tục duy trì những giá trị sản phẩm chất lượng tới tay khách hàng, nó là framework phục vụ mục đích giải quyết vấn đề để tạo nên sản phẩm. Trong khi đó Agile là một tuyên ngôn gồm những giá trị và quy tắc mà chúng ta nên tuân thủ theo khi tham gia vào một dự án phát triển phần mềm. Nhưng về Agile là nó là những lý thuyết rất lớn, chứ không phải là một mô hình để chúng ta áp dụng vào luôn quá trình phát triển phần mềm như Scrum. Việc sử dụng Scrum framework giúp chúng ta phát triển phần mềm, tư duy theo xu hướng Agile một cách dễ dàng hơn.

Trong ngành phát triển phần mềm, những yêu cầu, mục tiêu về sản phẩm luôn được thay đổi mỗi ngày. Scrum framework giúp ta dễ dàng thích nghi với những sự thay đổi đó.

Trong Scrum, chúng ta sẽ xây dựng phần mềm dựa trên những vòng lặp gọi là sprint. Sprint giúp chúng ta chia nhỏ những vấn đề lớn trở thành những chức năng nhỏ mà từ đó người phát triển phần mềm có thể vạch rõ kế hoạch để phát triển tính năng đó trong thời gian hữu hạn

Scrum hoạt động dựa trên những vòng lặp nhỏ liên tục (sprint) thường kéo dài 2 tuần. Lợi ích của việc này chính là chúng ta có thể dễ dàng đón nhận sự thay đổi, giảm thiểu rủi ro và chi phí, thu thập feedback trực tiếp từ khách hàng nhanh nhất có thể. Tăng tốc độ phát triển phần mềm và nhìn thấy thành quả sản phẩm sớm hơn. Ở cuối mỗi sprint sẽ là những buổi họp để đánh giá xem những gì đã hoàn thành và chưa hoàn thành của dự án. Từ đó giúp thành viên trong team luôn giữ được sự tập trung lâu dài với dự án.

### Quy trình vận hành Scrum trong phát triển phần mềm

Scrum gồm có:

* 3 gia trị cốt lõi:
* Minh bạch: đảm bảo rằng tất cả các khía cạnh quan trọng của tiến trình cần phải được hiển thị rõ ràng cho tất cả mọi người có liên quan đến việc phát triển sp

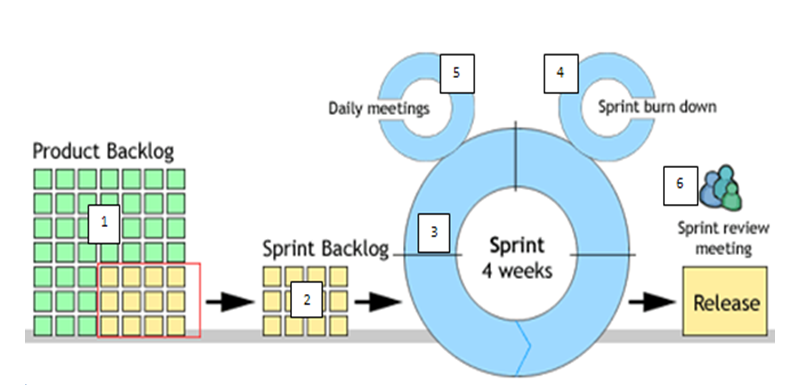
- Giúp giảm thiểu rủi ro

- Nâng cao chất lượng sản phẩm

- Gia tăng sự tin cậy

* Thanh tra: Người sd scrum phải thường xuyên thanh tra về các tạo tác, tiến độ để đảm bảo mục tiêu scrum, và phát hiện điểm bất thường ngoài ý muốn
* Thích nghi: Scrum là một trong những phương pháp phát triển rất linh hoạt. Nhờ đó nó mang lại tính thích nghi rất cao. Scrum có thể phản hồi lại các thay đổi một cách tích cực nhờ đó mang lại nhiều thành công lớn cho dự án
* 3 vai trò:
* **Product Owner (Chủ sản phẩm):**Là người chịu trách nhiệm về sự thành công của dự án, là người định nghĩa các yêu cầu và đánh giá cuối cùng đầu ra của các nhà phát triển phần mềm.
* **Scrum master (Đội trưởng):**Là người hiểu biết sâu sắc về Scrum và đảm bảo nhóm có thể làm việc hiệu quả với Scrum.
* **Team (Đội sản xuất):**Là một nhóm liên chức năng có nhiệm vụ giải quyết các yêu cầu từ Product Owner để tạo ra các gói sản phẩm tốt nhất.
* 4 cuộc họp:
* Sprint Planning (Lập kế hoạch cho Sprint)
* Daily Scrum (Buổi họp hằng ngày)
* Sprint Review (Buổi họp đánh giá)
* Sprint Retrospective (Buổi họp cải tiến)

Quy trình vận hành của Scrum được thực hiện thông qua các vòng lặp phát triển gọi là sprint. mỗi sprint kéo dài không quá 4 tuần và liên tiếp không bị gián đoạn, sprint này sẽ bắt đầu ngay sau khi sprint trc kết thúc



Khung làm việc của Scrum

Product Owner taọ ra Product backlog: gồm tập hợp tất cả các yêu cầu của 1 sản phẩm

Nhóm phát triển sẽ thực hiện hoá dần Product Backlog

Trước khi thực hiện sprint chúng ta sẽ tổ chức buổi họp Sprint Planning ( gồm có đội sản xuất và có cả đại diện của khách hàng) để lập kế hoạch cho từng Sprint ( tloi các câu hỏi của đội dự án, đội dự án sẽ thảo luận và đưa ra các cái task công việc cần làm trong 1 sprint).

Kết quả của buổi họp này là Sprint Backlog chưa các công việc cần làm trong 1 Sprint

Trong quá trình thực hiện Sprint, nhóm sẽ phải cập nhật Sprint Backlog và thực hiện cuộc họp hằng ngày (Daily Scrum) để chia sẻ tiến độ công việc, và các vướng mắc trong quá trình làm việc.

Kết thúc 1 sprint nhóm tạo ra các gói phần mềm có chức năng hoàn chỉnh, sẵn sàng chuyển giao cho khách hàng.Buổi họp Sprint Review ở cuối mỗi Sprint sẽ giúp khách hành thấy được nhóm đã chuyển giao những gì, và những gì phải làm hoặc phải thay đổi cải tiến

Sau khi kết thúc đánh giá Sprint, chúng ta sẽ tổ chức cuộc họp cải tiến Sprint (Sprint Retrospective) để cái tiến quy trình làm việc trước khi Sprint tiếp theo bắt đầu

Các sprint cứ tổ chức liên tục như vậy cho đến khi hoàn thành sản phẩm

### Lợi ích của quy trình Scrum

## Chất lượng tốt hơn

Các dự án tồn tại là để hoàn thành một mục tiêu hoặc tầm nhìn nào đó. Scrum cung cấp khung làm việc cho phép liên tục phản hồi và phát hiện sai sót nhằm đảm bảo chất lượng cao nhất có thể. Scrum giúp đảm bảo chất lượng bằng những phương pháp sau:

* Làm rõ và phát triển những yêu cầu đúng thời điểm để những hiểu biết về tính năng sản phẩm luôn được cập nhập.
* Kết hợp kiểm thử hằng ngày và phản hồi của Product Owner trong quá trình phát triển, giúp cho đội ngũ phát triển nhanh chóng phát hiện vấn đề.
* Cải thiện đầu ra (sản phẩm hoặc dịch vụ) của Nhóm Scrum thường xuyên và liên tục thông qua các buổi Sơ kết Sprint với các bên liên quan.
* Thực hiện các buổi Cải tiến Sprint nhằm giúp Nhóm Scrum liên tục cải tiến quy trình, công cụ, mối quan hệ và môi trường làm việc.
* Hoàn thành công việc sử dụng Định nghĩa Hoàn thành (DoD) cho các công đoạn phát triển, kiểm thử, tích hợp và tài liệu hóa.

## Giảm Thời gian Đưa ra Thị trường (time to market)

Scrum đã được chứng minh là đem giá trị đến người dùng cuối nhanh hơn phương pháp truyền thống từ 30-40%. Thời gian được rút ngắn nhờ những yếu tố sau:

* Quá trình phát triển diễn ra sớm hơn do phần xây dựng tài liệu lúc đầu của dự án mô hình waterfall (thường mất hàng tháng trời) đã được thay thế bằng vai trò Product Owner trong Nhóm Scrum nhằm thực hiện các yêu cầu đúng tiến độ và cung cấp giải trình theo thời gian thực.
* Những yêu cầu ưu tiên cao được tách biệt với yêu cầu ưu tiên thấp. Việc đưa giá trị tăng dần đến người dùng cuối đồng nghĩa với việc những yêu cầu giá trị cao nhất (đồng thời rủi ro cũng cao nhất) sẽ được thực hiện trước những yêu cầu có giá trị và độ rủi ro thấp hơn. Không cần thiết phải chờ đến khi hoàn thành toàn bộ dự án mới tung sản phẩm/dịch vụ ra thị trường.
* Các chức năng hoàn chỉnh được đóng gói trong mỗi Sprint. Cuối mỗi Sprint, Nhóm Scrum cho ra gói tăng trưởng của sản phẩm/dịch vụ có khả năng chuyển giao được.

## Lợi nhuận từ đầu tư (ROI) cao hơn

Thời gian ra thị trường ngắn hơn là lý do chính khiến những dự án Scrum có lợi nhuận từ đầu tư (ROI) cao hơn. Thời gian ra thị trường sớm hơn dẫn tới doanh thu, các lợi ích khác sẽ có sớm hơn. Kết quả là chúng ta tích lũy sớm hơn, do vậy tổng lợi tức thu lại cao hơn theo thời gian. Đây là nguyên lý cơ bản trong tính toán giá trị hiện tại thuần (NPV – Net Present Value). Bên cạnh lợi ích về thời gian tung ra thị trường sớm, ROI của các dự án Scrum còn tăng bởi:

* Phản hồi thường xuyên thông qua các buổi Sơ kết Sprint trực tiếp từ cổ đông bao gồm khách hàng giúp cho việc sửa lỗi sai diễn ra kịp thời, qua đó giúp tiết kiệm thời gian và chi phí hơn.
* Nhờ tự động hoá và kiểm thử từ sớm nên giảm được các công việc thừa thãi, và triển khai sản phẩm nhanh hơn.
* Giảm thiểu chi phí thất bại. Nếu một dự án Scrum thất bại, nó thất bại sớm hơn và nhanh hơn so với dự án Waterfall.

## Khách hàng hài lòng hơn

Nhóm Scrum cam kết mang đến sản phẩm và dịch vụ khiến khách hàng hài lòng. Scrum cho phép các nhà tài trợ dự án trở nên vui vẻ hơn thông qua:

* Hợp tác với khách hàng như những cộng sự và đảm bảo sự liên kết và tham gia của khách hàng trong suốt dự án.
* Có một Product Owner là chuyên gia về yêu cầu của sản phẩm và nhu cầu của khách hàng.
* Giữ cho Product Backlog luôn được cập nhật và được ưu tiên hóa để phản ứng nhanh chóng với những thay đổi.
* Trình diễn các chức năng\tính năng hoạt động tốt với cổ đông nội bộ và khách hàng vào mỗi kì Sơ kết Sprint.
* Đưa sản phẩm đến người dùng cuối nhanh và thường xuyên hơn với nhiều lần phát hành hơn thay vì đưa ra toàn bộ sản phẩm vào phút cuối.
* Nguồn vốn cho dự án tăng dần thay vì đòi hỏi các khoản cam kết lớn ngay lúc đầu

## Tinh thần nhóm cao hơn

Làm việc với những người hạnh phúc và yêu thích công việc của mình có thể rất thoả mãn và vui vẻ. Tự quản khiến cho việc ra quyết định vốn là của nhà quản lý giờ đây cũng là của thành viên Nhóm Scrum. Scrum cải thiện tinh thần của các thành viên nhóm bằng những cách sau:

* Là một phần của nhóm tự quản và tự tổ chức khiến cho thành viên trở nên sáng tạo, đột phá và được ghi nhận chuyên môn nhiều hơn.
* Các nhóm phát triển có thể tổ chức cấu trúc nhóm làm việc tùy theo các phong cách làm việc và cá tính khác nhau.
* Nhóm Scrum có thể đưa ra các quyết định phù hợp dựa trên sự cân bằng giữa công việc và đời sống cá nhân thành viên nhóm.
* Mối quan hệ ngang bằng với đại diện kinh doanh (Product Owner) ở cùng một đội giúp cân bằng các ưu tiên kinh doanh và kỹ thuật và phá bỏ rào cản trong tổ chức.
* ScrumMaster, với vai trò hỗ trợ Nhóm Scrum sẽ xoá bỏ đi những rào cản và bảo vệ Nhóm Phát triển khỏi những can thiệp từ bên ngoài.
* Tập trung vào các hoạt động bền vững đảm bảo mọi người không cảm thấy kiệt sức vì stress hoặc quá nhiều việc.
* Làm việc liên chức năng giúp tạo điều kiện cho thành viên Nhóm Phát triển học được kỹ năng mới và phát triển hơn khi hướng dẫn người khác.
* Khuyến khích cách tiếp cận lãnh đạo-phục vụ (servant- leader) nhằm hỗ trợ các thành viên trong Nhóm Scrum trong việc tự quản và chủ động tránh các phương pháp ra lệnh và kiểm soát.
* Tạo một môi trường hỗ trợ và tin tưởng lẫn nhau giúp tăng động lực và tinh thần nhân viên.
* Có những cuộc đối thoại mặt đối mặt giúp hạn chế những ảnh hưởng của giao tiếp không rõ ràng.
* Cuối cùng, Nhóm Scrum có thể cùng đồng thuận về những quy tắc để hoàn thành công việc.

## Gia tăng việc cộng tác và sự sở hữu

Khi Nhóm Scrum nhận trách nhiệm về một dự án hay sản phẩm nào đó, họ có thể đem lại những kết quả tuyệt vời. Nhóm Scrum hợp tác và quản lý chất lượng, quản lý hiệu quả dự án qua những cách sau:

* Xây dựng Nhóm Phát triển, Product Owner và ScrumMaster làm việc sát sao cùng nhau hằng ngày.
* Thực hiện các buổi Lập kế hoạch Sprint, cho phép Nhóm Phát triển tổ chức công việc của mình xoay quanh các ưu tiên về kinh doanh.
* Có các buổi Scrum Hằng ngày để Nhóm Phát triển có thể biết được các công việc đã hoàn thành, công việc cần làm và những khó khăn cản trở là gì.
* Thực hiện các buổi Sơ kết Sprint để Product Owner có thể vạch ra những quyết định ưu tiên của mình và Nhóm Phát triển có thể trình bày và thảo luận về sản phẩm trực tiếp với các bên liên quan.
* Thực hiện những buổi Cải tiến Sprint giúp cho thành viên Nhóm Scrum có thể rà soát lại công việc trước đây và đề xuất ra những giải pháp tốt hơn với mỗi Sprint.
* Làm việc trong môi trường mở giúp cho việc giao tiếp và phối hợp giữa các thành viên nhóm, Product Owner và ScrumMaster diễn ra nhanh hơn.
* Ra quyết định với sự đồng thuận cao.

## Chuẩn đánh giá chính xác hơn

Số liệu được Nhóm Scrum dùng để tính toán thời gian và chi phí, đo lường hiệu suất dự án và đưa ra các quyết định thường có liên quan và chuẩn xác hơn so với số liệu của các dự án truyền thống. Số liệu phù hợp hơn trong những dự án Scrum là vì:

* Không ai khác ngoài những người thực hiện dự án sẽ phải đánh giá nguồn lực mà dự án cần.
* Khung thời gian và ngân sách dựa trên sự phát triển thực tế của hiệu suất và khả năng của Nhóm Phát triển.
* Sử dụng những ước tính tương đối hơn là ước tính số giờ hoặc ngày để điều chỉnh các nỗ lực phù hợp hơn với khả năng và kiến thức của một Nhóm Phát triển.
* Chưa đến một phút mỗi ngày, lập trình viên có thể cập nhật biểu đồ burndown, cung cấp cái nhìn trực quan về tiến độ tiến tới mục tiêu Sprint của cả Nhóm Phát triển
* Cuối mỗi Sprint, Product Owner có thể so sánh chi phí thực của dự án (actual cost – AC) cùng chi phí cơ hội của các dự án trong tương lai (opportunity cost – OC) với giá trị mà dự án đang thực hiện lúc này mang lại (V) để biết được khi nào thì nên chấm dứt một dự án và bắt đầu một cái mới. Bạn không cần thiết phải chờ đến khi kết thúc dự án mới biết được giá trị của nó là gì.

## Cải thiện sự minh bạch của tiến độ dự án

Trong dự án Scrum, mỗi thành viên của đội dự án (bao gồm cả Nhóm Scrum và bên liên quan) có cơ hội tìm hiểu dự án đang được triển khai ra sao vào bất cứ thời điểm nào. Tính minh bạch và rõ ràng khiến cho Scrum trở thành một mô hình phù hợp cho các nhóm dự án tìm ra vấn đề và dự đoán tiến độ, hướng đi của dự án một cách chính xác hơn. Dự án Scrum cung cấp một cái nhìn rõ ràng về tiến độ bởi:

* Đề cao sự giao tiếp chân thật, cởi mở giữa Nhóm Scrum, các bên liên quan, khách hàng và bất cứ ai nằm trong tổ chức muốn biết về dự án.
* Scrum Hằng ngày cung cấp cái nhìn thường xuyên về tiến độ cũng như khó khăn của Nhóm Phát triển.
* Scrum Hằng ngày xung quanh bảng công việc giúp cho lập trình viên tự tổ chức và xác định công việc có độ ưu tiên cao nhất trong ngày.
* Sử dụng thông tin từ những buổi Scrum Hằng ngày, biểu đồ Sprint burn-down và bảng công việc giúp cho đội dự án theo dõi được tiến độ của từng Sprint.
* Buổi Cải tiến Sprint giúp cho thành viên Nhóm Scrum xác định được cái gì hiệu quả và cái gì không nên làm để đưa ra kế hoạch để cải tiến.
* Trình diễn kết quả của Sprint trong những buổi Sơ kết Sprint. Bất kì ai trong tổ chức đều có thể tham gia một buổi Sơ kết Sprint kể cả khi người đó là thành viên của Nhóm Scrum khác.

## Gia tăng sự kiểm soát với dự án

Nhóm Scrum có cơ hội lớn để kiểm soát năng suất dự án và chỉnh sửa cần thiết vì những yếu tố sau:

* Điều chỉnh ưu tiên xuyên suốt dự án tại mỗi nhịp sprint thay vì những mốc lớn giúp cho tổ chức có được dự án với thời gian và chi phí cố đinh trong khi phí ở có thể thay đổi.
* Khuyến khích sự thay đổi khiến cho đội dự án có thể phản ứng với những yếu tố bên ngoài như nhu cầu thị trường.
* Sự tương tác trong buổi Scrum Hằng ngày giúp Nhóm Scrum nhanh chóng tìm ra vấn đề và cùng nhau hoàn thành các yêu cầu.
* Sự cập nhật hằng ngày trên Sprint Backlog cho phép các biểu đồ Sprint burn-down thể hiện tiến độ của Sprint một cách chính xác, tạo điều kiện cho Nhóm Scrum thay đổi vào thời điểm họ thấy có vấn đề xảy ra.
* Giao tiếp mặt đối mặt giúp xóa bỏ rào cản giao tiếp và tìm ra các giải pháp.
* Buổi Sơ kết Sprint giúp cho các bên liên quan thấy sản phẩm đang được thực hiện, có thể đưa ra những phản hồi cho Product Owner cần dùng để dự án đi đúng hướng.
* Buổi Cải tiến Sprint giúp cho Nhóm Scrum đưa ra những điều chỉnh vào cuối mỗi Sprint để nâng cao chất lượng sản phẩm, tăng năng suất Nhóm Phát triển và làm mịn tiến độ dự án.

Sẽ có rất nhiều cơ hội để thanh tra và thích nghi trong các dự án Scrum giúp cho tất cả thành viên đội dự án – Nhóm Phát triển, Product Owner, ScrumMaster và các bên liên quan – thực hiện việc điều chỉnh và tạo ra các sản phẩm tốt đẹp hơn.

## Giảm thiểu rủi ro

Scrum giúp giảm thiểu rủi ro dự án thất bại hoàn toàn – đổ một đống tiền và thời gian nhưng không thu lại lợi nhuận – bằng cách đưa ra sản phẩm hữu hình ngay từ sớm và buộc Nhóm Scrum phải thất bại từ sớm nếu họ chắc rằng sẽ thất bại thông qua những yếu tố sau:

* Hoàn thành những hạng mục backlog có độ rủi ro cao nhất trước tiên tạo ra lộ trình công việc dài nhất thông qua các vấn đề, thất bại sớm đồng thời ít tốn kém nhất.
* Triển khai dự án theo các Sprint, đảm bảo có một quãng thời gian ngắn giữa đầu tư dự án ban đầu và việc sản phẩm đó sẽ thất bại nhanh chóng hoặc là cách tiếp cận này có hiệu quả.
* Ở ngay những Sprint đầu tiên, ta thấy sản phẩm dần được hình thành nên ngay cả khi dự án bị hủy bỏ, với những yêu cầu có giá trị cao nhất và những yêu cầu rủi ro cao nhất đã được phát triển và có thể chuyển giao khách hàng nếu muốn.
* Phát triển các yêu cầu về định nghĩa hoàn thành ở mỗi Sprint thứ mà các nhà tài trợ dự án tài trợ dự án đã đưa ra, những tính năng hữu dụng, hay bất cứ điều gì có thể diễn ra trong tương lai.
* Cung cấp phản hồi liên tục về sản phẩm và tiến độ.

## So sánh giữa mô hình Agile/Scrum với mô hình truyền thống

Những mô hình phát triển phần mềm theo hướng truyền thống có nhiều loại khác nhau như: Waterfall, V model,.. Nhưng mô hình điển hình cho mô hình phát triển phần mềm truyền thống nhất

| **Tiêu chí** | **Agile** | **Waterfall** |
| --- | --- | --- |
| Khái niệm | Là mô hình phát triển phần mềm linh hoạt, lặp lại liên tục của quá trình phát triển và thử nghiệm trong quá trình phát triển phần mềm. [3] | Là mô hình phát triển theo một trật tự cố định và nhóm phát triển dự án không chuyển sang giai đoạn phát triển hoặc thử nghiệm tiếp theo cho đến khi bước trước đó được hoàn thành thành công. |
| Đặc điểm | - Sự lặp lại của các giai đoạn trong suốt dự án.  - Cho phép thực hiện các thay đổi đối với các yêu cầu ngay cả khi đã hoàn thành kế hoạch ban đầu.  - Hoạt động đặc biệt tốt với Time & Materials(nguồn lực) hoặc tài trợ không cố định.  - Cung cấp các sản phẩm, tính năng nhanh chóng cho người dùng, sau mỗi lần lặp (Sprint) | - Mỗi giai đoạn chỉ xảy ra một lần trong suốt dự án.  - Không có chỗ để thay đổi các yêu cầu khi bắt đầu phát triển dự án.  - Hoạt động tốt, giảm rủi ro trong các hợp đồng giá cố định.  - Chỉ cho phép cung cấp sản phẩm cho người dùng sau khi đã hoàn thành toàn bộ các quy trình. |
| Tài liệu | Không bắt buộc phải có tài liệu, tuy nhiên nếu đó là một phương thức truyền đạt thông tin hiệu quả thì vẫn nên tạo tài liệu. | Tài liệu cần được tạo cho mỗi quy trình và được kết nối với quy trình tiếp theo. Ngoài ra, tài liệu cũng có vai trò như là bằng chứng (evidence) của dự án sau khi toàn bộ quá trình hoàn tất. |
| Nguồn lực | - Các thành viên trong nhóm có thể hoán đổi vị trí cho nhau. | - Bắt buộc phải có một người phụ trách từng quy trình. |
| Tần suất kiểm thử | Thực hiện thử nghiệm cùng lúc với lập trình hoặc ít nhất là trong cùng một lần lặp lại với lập trình. | Giai đoạn “Thử nghiệm” xuất hiện sau giai đoạn “Xây dựng” |
| Quy mô dự án | Các dự án có yêu cầu chưa rõ ràng, được dự kiến thay đổi và phát triển. | Phù hợp cho các dự án có yêu cầu nhất định, không xuất hiện các thay đổi về yêu cầu phát triển. |
| Hướng tiếp cận | Tập trung vào sự hài lòng của người dùng và do đó sự tham gia của người dùng trong giai đoạn phát triển. | Một quá trình nội bộ và không yêu cầu sự tham gia của người dùng |
| Cải tiến quy trình | Có thể cải thiện quy trình cho mỗi lần lặp; nếu lần này không ổn, hoàn toàn có thể cải thiện ở những lần tiếp theo. | Chúng ta chỉ có thể nhìn ra được những điểm cần cải thiện sau khi dự án hoàn thành |

## Áp dụng mô hình Agile/Scrum vào tổ chức, doanh nghiệp(Hiếu)

1. **Các vấn đề khi đưa Scrum vào tổ chức**

- Khi quyết định mang Scrum về tổ chức, chúng ta sẽ phải giải quyết một loạt các vấn đề để tổ chức có thể đón nhận Scrum thành công và gặt hái những lợi ích do Scrum mang lại. Những vấn đề nổi cộm có thể dễ dàng nhìn ra như: thay đổi các vai trò công việc của cá nhân và phòng ban hiện có, thay đổi cách thức đánh giá kết quả, và vẽ lại lộ trình nghề nghiệp cho hàng loạt vị trí cũ và mới.

* Scrum sẽ xung đột vai trò\tổ chức: có thể dẫn đến việc phải định nghĩa lại công việc của quản trị dự án do cách thức quản trị dự án đã thay đổi, sắp xếp lại chức năng và nhiệm vụ của những phòng ban chức năng để tổ chức những nhóm Scrum liên chức năng, những phòng ban truyền thống như phòng đảm bảo chất lượng, các phòng kiểm thử, phòng thiết kế ..., có thể sẽ biến mất khỏi cơ cấu tổ chức, hoặc sẽ phải định vị lại chức năng và nhiệm vụ.
* Song song với việc thay đổi các vị trí, phòng ban, tổ chức cũng sẽ phải thay đổi lại cách thức đánh giá hiệu suất, định nghĩa lại sự thành công để vừa đảm bảo cá nhân vẫn nỗ lực, nhưng lại khuyến khích tinh thần hợp tác; vừa đánh giá đúng giá trị của cá nhân, lại vừa đánh giá thành tích dựa trên sự chung sức trong tập thể. Đây vừa là vấn đề kĩ thuật, nhưng cũng là vấn đề mang tính giá trị quan cần sự thay đổi sâu sắc trong tư duy của lãnh đạo.
* Khi đưa Scrum vào, hàng loạt vai trò mới phát sinh như ScrumMaster, Product Owner, Agile Coach, Scrum Coach, và những công việc của từng thành viên trong đội phát triển sẽ thay đổi về chất. Điều này đòi hỏi bộ phận quản lí nhân sự và lãnh đạo phải quy hoạch lại lộ trình nghề nghiệp, hệ thống mô tả nghiệp vụ và những chính sách đi kèm để phát triển đội ngũ vững mạnh đáp ứng nhiệm vụ theo cách làm mới mà Scrum mang lại. Không chú trọng tới yếu tố này thì những cách áp dụng Scrum chỉ mang tính bề mặt, thiếu bền vững và khó lòng gặt hái được thành công lâu dài.
* Cuối cùng, việc du nhập một cách làm mới, một công cụ mới hay một công nghệ mới về bản chất là một sự thay đổi, lãnh đạo công ty sẽ phải lập kế hoạch quản lí sự thay đổi để đảm bảo sự du nhập Scrum thành công, vượt qua những trở ngại và phản đối để có những thay đổi căn bản, cắm rễ lâu bền vào văn hóa của công ty.

1. **Các tình huống áp dụng Scrum**

- Ở mức độ thấp nhất, Scrum được dùng với mục đích thí điểm. Khi doanh nghiệp chưa sẵn sàng để dùng Scrum cho các dự án quan trọng, thì mô hình này tỏ ra phù hợp. Theo đó Nhóm Scrum được lập ra và thực hiện một nhiệm vụ (thật hoặc giả lập) ngắn hạn chừng vài Sprint để kiểm tra xem Scrum có phù hợp với doanh nghiệp mình không, đội ngũ có dễ dàng làm chủ Scrum không, và có những vấn đề phát sinh nào. Nhiều doanh nghiệp chọn bước thí điểm là bước khởi đầu trước khi áp dụng đồng loạt Scrum cho toàn bộ tổ chức.

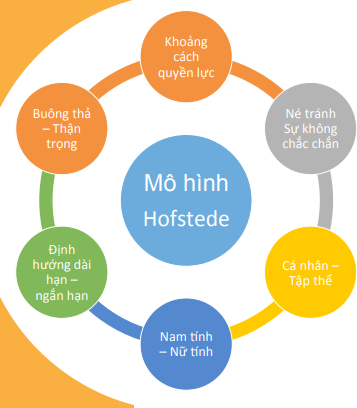
- Mức độ phổ biến nhất là các doanh nghiệp sử dụng Scrum như là một khuôn khổ mới cho quản trị dự án. Theo đó, tùy theo tính trọng yếu mà một số dự án sẽ được tổ chức dưới dạng Scrum. Đây là hình thức đầu tư ngắn hạn, thu lợi nhanh, phù hợp với các công việc thời vụ hoặc do doanh nghiệp chưa có chiến lược sử dụng Scrum lâu dài.

- Mức độ cao hơn của việc vận dụng Scrum là dùng nó làm quy trình tiêu chuẩn để sản xuất phần mềm. Trong trường hợp này toàn bộ đội ngũ sản xuất được tổ chức dưới dạng các nhóm Scrum, nếu sản phẩm lớn hơn sẽ áp dụng những cách vận dụng Scrum ở quy mô lớn (tham khảo Chương 9) để tổ chức sản xuất. Đây là cách làm đòi hỏi chiến lược lâu dài, đầu tư bài bản cho con người, lộ trình nghề nghiệp, hạ tầng công nghệ và cơ sở vật chất đi kèm.

- Trong trường hợp bên trên, Scrum vẫn chỉ đóng vai trò thu hẹp trong đội ngũ phát triển sản phẩm. Trường hợp sử dụng Scrum với phạm vi lớn hơn là khi toàn bộ tổ chức tạo lập được văn hoá Scrum để cùng hoạt động. Tổ chức sẽ lấy các giá trị Scrum làm giá trị căn bản của văn hoá công ty, các lãnh đạo sẽ vận dụng tư duy Agile và các quy trình Scrum cho các lĩnh vực vận hành, phát triển sản phẩm, và dịch vụ. Thông thường, các công ty nhỏ hơn (như các startup vài người đến vài chục người) cho đến những công ty cỡ vừa sẽ dễ dàng vận dụng Scrum theo cách này. Các doanh nghiệp lớn hơn sẽ đòi hỏi một chiến lược chuyển đổi bài bản cùng với các kế hoạch quản trị thay đổi đường dài.

1. **Các khía cạnh văn hoá cần quan tâm**

* Một trong những lo ngại rất phổ biến ở những nơi chuẩn bị du nhập Scrum là lo ngại những đặc thù văn hoá bản địa sẽ không phù hợp với văn hoá làm việc của Scrum vốn đánh giá cao giá trị tự chủ, làm việc nhóm, hướng đến khách hàng và sẵn sàng thay đổi để thích nghi.
* Chúng ta có thể vận dụng mô hình do Tiến sĩ Hofstede đề xuất để hiểu thêm về đặc thù văn hoá quốc gia và có cách để vượt qua những trở ngại thuộc phạm trù văn hoá. Tất nhiên mô hình này thật khó mà đầy đủ, nhưng trong chừng mực nào đó, nó hữu ích để việc du nhập Scrum vào một tổ chức mới thuận lợi hơn.

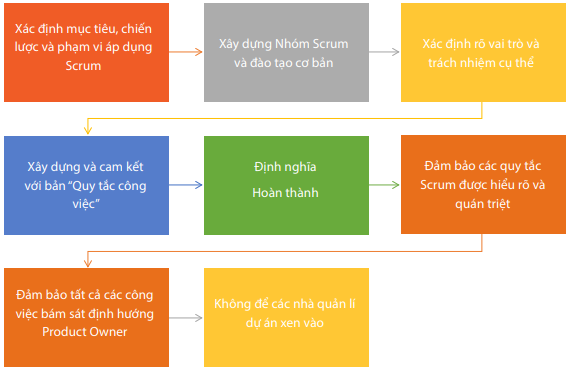


1. **Quản trị dự án linh hoạt với Scrum**

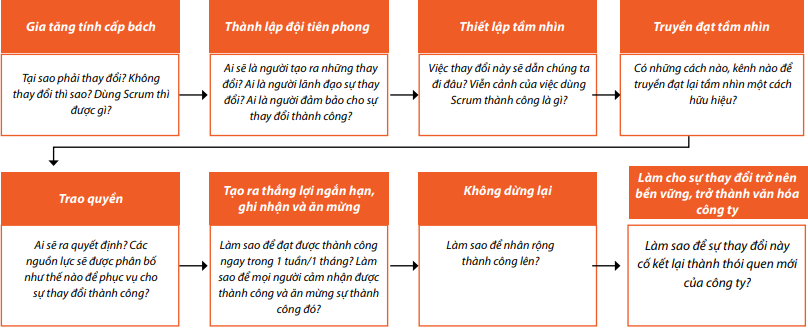
* Quản trị dự án là một trong các tình huống phổ biến nhất khi một doanh nghiệp bắt đầu vận dụng Scrum. Lợi ích của cách làm này là hạn chế sự ảnh hưởng quá lớn tới cơ cấu tổ chức và văn hoá công ty trong khi vẫn đạt được những lợi ích mà Scrum mang lại như khả năng chuyển giao nhanh chóng, thích ứng nhanh, cải tiến liên tục quy trình làm việc và đảm bảo chất lượng cao.
* Việc vận dụng Scrum vào quản trị dự án có thể được tiến hành khá thuận lợi với việc định nghĩa lại các vai trò trong nhóm dự án và chuyển đổi quy trình dự án sang các Sprint. Điều cần lưu tâm nhất có lẽ là nằm ở việc định nghĩa lại vai trò của nhà quản trị dự án. Nhiều người nhầm lẫn vai trò của ScrumMaster với Project Manager, và gọi ScrumMaster là một “project manager” kiểu mới. Thực tế khác hẳn. Công việc của Project Manager truyền thống đã được phân bố lại cho các vai trò của một Nhóm Scrum để nhóm tự quản lí lấy công việc của mình. Do vậy đối với các công việc được định nghĩa trong Scrum cần đảm bảo, tránh ScrumBut. Người Project Manager hiện có của nhóm có thể đóng vai trò mới (Product Owner hoặc ScrumMaster) nhưng không được vượt qua các quy tắc mà Scrum đã thiết lập.
* Bên cạnh việc định nghĩa lại quy trình và các vai trò, quản trị dự án linh hoạt (hoặc ScrumMaster) cần thiết lập được một hệ thống báo cáo (dùng các công cụ Backlog hoặc Project Dashboard) phù hợp để luôn đảm bảo thông tin minh bạch đối với các cấp quản lí và các bên liên quan.

1. **Lộ trình áp dụng Scrum**

- Lộ trình đơn giản để dưa Scrum vào tổ chức:



1. **Quản lí sự thay đổi**

* Thay đổi cung cách làm việc là một thay đổi căn cơ, đòi hỏi sự lưu tâm của lãnh đạo. Phạm vi áp dụng càng lớn thì việc quản lí thay đổi càng quan trọng. Chúng ta có thể vận dụng mô hình quản trị sự thay đổi của John Kotter để quản lí sự thay đổi khi tiếp nhận Scrum vào tổ chức. Bạn cần có một kế hoạch thay đổi chu toàn, triển khai và giám sát, đánh giá liên tục và thích nghi để hướng tới mục tiêu định trước. Một kế hoạch thay đổi có thể được tổ chức dưới dạng dự án gồm một Nhóm Scrum, với “Phần tăng trưởng” chính là những thay đổi bạn muốn có ở trong tổ chức
* Quy trình trong quản lý sự thay đổi:

## Tài liệu tham khảo

[1]<https://www.scrum.org/resources/scrum-glossary>

[2]<https://www.scrum.org/resources/what-is-scrum>

[3]<https://www.javatpoint.com/software-engineering-agile-model>

[4]<https://www.atlassian.com/agile/>

[5]<https://vn.got-it.ai/blog/5-phuong-phap-agile-pho-bien-trong-phat-trien-phan-mem-ma-ban-nen-biet>

[6]https://iviettech.vn/blog/1783-scrum-best-practice-thuc-hanh-ap-dung-scrum.html

[7]https://hocvienagile.com/wp-content/uploads/2017/02/Cam-nang-Scrum\_Chuong-6.pdf